Patent **Attorney Docket No.: Intel 2207/17039**
Assignee: Intel Corporation

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANTS	:	Per H. HAMMARLUND et al.
SERIAL NO.	:	10/749,271
FILED	:	December 30, 2003
TITLE	:	METHOD AND APPARATUS FOR ENABLING AN ADAPTIVE REPLAY LOOP IN A PROCESSOR
GROUP ART UNIT	:	2183
EXAMINER	:	Jacob Andrew PETRANEK

M/S: APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

REPLY BRIEF

Dear Sir:

This is in reply to issues raised by the Examiner in his Answer of August 4, 2008.

REMARKS/ARGUMENTS

Applicants respectfully submit the arguments in the Examiner's Answer are incorrect for at least the following reasons.

Applicants maintain the cited references do not teach, suggest or describe at least "an adaptive replay system comprising … *a selector device* coupled to said staging area *to place said instruction in an optimal position within said replay loop…*" (*e.g.*, as described in claim 1).

For example, the section of Merchant describing the cited element 150 fails to teach or suggest the relevant limitations for reasons similar to those described in the Appeal Brief. As discussed element 150 is described as a checker, while element 154 is described as a replay queue loading controller. The cited section states:

> If the checker 150 determines that the instruction has not executed properly, the instruction will then be returned to multiplexer 116 to be replayed (*i.e.*, re-executed). Each instruction to be replayed will be returned to mux 116 via one of two paths. Specifically, if the checker 150 determines that the instruction should be replayed, the Replay Queue Loading Controller 154 determines whether the instruction should be sent through a replay loop 156 including staging queues E and F, or whether the instruction should be temporarily stored in a replay queue 170 before returning to mux 116.
>
> Instructions routed via the replay loop 156 are coupled to mux 116 via line 161. Instructions can also be routed by controller 154 for temporary storage in replay queue 170 (prior to replay). The instructions stored in replay queue 170 are output or unloaded under control of replay queue unloading controller 179. The instructions output from replay queue 170 are coupled to mux 116 via line 171. The operation of replay queue 170, Replay Queue Loading Controller 154 and Replay Queue Unloading Controller 179 are described in detail below.

The first sentence of the cited section introduces the "re-execut[ion]" process. Specifically, if the checker 150 determines the instruction did not execute properly, it is returned to the multiplexer 116. To do this, a conditional determination is made which decides to send the instruction down one of two paths. The first outcome of this conditional determination sends

the instruction to the replay loop 156, while the second outcome sends it temporarily to the replay queue 170, directed by the cited controller 154, before sending it to the mux 116. Instructions stored in replay queue are sent to controller 179, and are coupled to mux 116 (described above) by line 171. Element 154 does not place the instruction in an optimal position of the replay loop, but instead it merely directs the instruction to queue 170 for temporary storage before replay. This is not the same as optimally rearranging the order of instructions (…"an adaptive replay system comprising … a selector device coupled to said staging area to place said instruction in an optimal position within said replay loop…" – as described in, for example, claim 1) at all.

Applicants submit at the heart of the cited section is this two-outcome conditional determination of where to send an instruction that hasn't executed properly. However, making a conditional determination upon which one of two steps may be followed is not the same as placing an instruction in an optimal position within a replay queue (as described in embodiments of the present application) at all. Indeed, the cited section does not discuss placement of an instruction for any reason anywhere (other than re-sending the instruction as discussed above).

The Examiner further argues element 150 determines if a replay is necessary and element 154 determines which replay path will be used for the instruction. Therefore, it asserts, element 154 places the instruction in an optimal position for the processor by either sending it through the staging queues E and F, or sending it to the replay queue where it could be delayed.

Applicants maintain this assertion fails to support a proper rejection for reasons similar

to those discussed above. Specifically, placing an instruction in an optimal position for the processor is not the same as placing an instruction in an optimal position within said replay loop (*e.g.*, as described in embodiment of the present application). As discussed above, the cited reference makes a conditional determination entailing either returning an instruction to the replay loop 156 to be repeated or sending it temporarily to the replay queue 170 before sending it to the mux 116. Such a determination or operation entails either repeating the instruction as a whole, or sending it temporarily to a replay queue 170. It does not entail placing an instruction in an optimal position within said replay loop (e.g., as described in embodiment of the present application).

The Examiner offers column 6, lines 43-49 for support. However, the cited section merely describes controller 154 in a way similar to that discussed above. Specifically, it describes controller 154 directs the instruction to queue 170 for temporary storage before replay. This is not the same as the relevant limitations described in claim 1, as discussed above.

The Examiner finally cites to column 7, lines 23-25 and lines 43-63. The first section, column 7, lines 23-25, only generally describes the possibility of long latency instructions that may require many iterations through replay loop 156. However, it does not describe anything relating to the cited element 154, or the relevant limitations of claim 1, anywhere.

The second section, lines 43-63, describes the process during a long-latency instruction processing sequence. The cited section describes the iterations through the replay loop 156. However, it does not describe anything relating to the cited element 154, or the relevant limitations of claim 1, anywhere.

Applicants maintain in order to support a proper rejection, the cited reference must describe at least placing an instruction in an optimal position within a replay loop. Since the for at least the reasons described above and in the Appeal Brief, the Merchant reference fails to do so, the current rejections of claims 1-30 are lacking and should be withdrawn.

Appellants therefore respectfully request that the Board of Patent Appeals and Interferences reverse the Examiner's decision rejecting claims 1-30 and direct the Examiner to pass the case to issue.

The Examiner is hereby authorized to charge any additional fees which may be necessary for consideration of this paper to Kenyon & Kenyon LLP Deposit Account No. **11-0600**.

Respectfully submitted,
KENYON & KENYON LLP

Date: October 6, 2008

By: /Sumit Bhattacharya/
Sumit Bhattacharya
(Reg. No. 51,469)
Attorneys for Intel Corporation

KENYON & KENYON LLP
333 West San Carlos St., Suite 600
San Jose, CA 95110

Telephone: (408) 975-7500
Facsimile: (408) 975-7501